PCT/EP2004/052200

Europäisches Patentamt
European Patent Office
Office européen des brevets

16.09.2004

Bescheinigung Certificate Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. Patent application No. Demande de brevet n°

03103545.4

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

R C van Dijk

Anmeldung Nr:
Application no.: 03103545.4
Demande no:

Anmeldetag:
Date of filing: 24.09.03
Date de dépôt:

Anmelder/Applicant(s)/Demandeur(s):

International Business Machines Corporation
New Orchard Road
Armonk, N.Y. 10504
ETATS-UNIS D'AMERIQUE

Bezeichnung der Erfindung/Title of the invention/Titre de l'invention:
(Falls die Bezeichnung der Erfindung nicht angegeben ist, siehe Beschreibung.
If no title is shown please refer to the description.
Si aucun titre n'est indiqué se referer à la description.)

Personalized web service description

In Anspruch genommene Priorität(en) / Priority(ies) claimed /Priorité(s) revendiquée(s)
Staat/Tag/Aktenzeichen/State/Date/File no./Pays/Date/Numéro de dépôt:

Internationale Patentklassifikation/International Patent Classification/
Classification internationale des brevets:

G06F17/30

Am Anmeldetag benannte Vertragstaaten/Contracting states designated at date of filing/Etats contractants désignées lors du dépôt:

AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LU MC NL PT RO SE SI SK TR LI

D E S C R I P T I O N

Personalized Web Service Description

1. BACKGROUND OF THE INVENTION

1.1. FIELD OF THE INVENTION

The present invention refers to the field of networked computer telecommunication, and in particular to a method and system for processing context information in web based applications for providing Web Services.

1.2. DESCRIPTION AND DISADVANTAGES OF PRIOR ART

Web services define a technique for describing software components to be accessed, methods for accessing these components, and discovery methods that enable the identification of relevant service providers. Web services are programming language-, programming model-, and system software neutral. In this regard, two prior art Web services standards are relevant. They are shortly sketched out and commented as follows in order to introduce into the problems concerned in prior art:

First, the *Simple Object Access Protocol (SOAP)* provides a means of messaging between a service provider and a service requestor. SOAP is independent of the underlying transport protocol, SOAP payloads can be carried on HTTP, FTP, JMS and other protocols.

Figure 1 gives a SOAP example carried by the HTTP POST request.

HTTP messages consist of requests from client to server and responses from server to client. Both types of message (Request and Response messages) consist of a start-line, zero or more header fields (also known as "headers"), an empty line indicating the end of the header fields, and possibly a message-body.

The structure of a HTTP request message is depicted in **Fig. 2:** the first line of that message specifies the method to be applied to the resource, the identifier of the resource, and the HTTP protocol version in use. The HTTP protocol defines multiple request methods like GET, HEAD, POST, PUT, DELETE, TRACE, CONNECT, OPTIONS. The method indicates the operation to be performed on the resource.

The resource upon which to apply the request is identified by a Request-URI, which is a Uniform Resource Identifier. Uniform Resource Identifiers are simply formatted strings, which identify--via name, location, address or any other characteristic--a resource. For example: The well-known HTTP URL scheme used to locate network resources via the HTTP protocol contains resource-URIs. The scheme specific syntax and semantics for http URLs are

http_URL = "http:" "//" host [":" port] [request-uri]

If the port is empty or not given, port 80 is assumed. The semantics are that the identified resource is located at the server listening for TCP connections on that port of that host, and the Request-URI identifies the resource. The syntax and semantics for Request-URI are

Request-uri = abs_path ["?" query-string]

where the abs_path is an identifier of the resource and the query string is any kind of information which can be used for processing the request.

The header fields carry meta-information associated with the request or response.

The message-body of an HTTP message is used to carry the entity-body associated with the request or response.

The message body depicted in Fig. 2 contains the actual SOAP message, which has a structure as given in **fig. 3**: Inside An envelope section a number of header fields 1,..n are defined, which construct the so called SOAP header, which is followed by by the actual SOAP body, which comprises a second number of so called "Body fields 1,..n.

Thus, the overall structure of a SOAP message carried over a network e.g. by a transport protocol like HTTP is depicted as a conglomeration of figs 1 to 3 in **fig 4**.

Figure 1 gives a SOAP example carried by the HTTP POST command. The HTTP request method is POST, the resource-URI is "/StockQuote", which is an absolute path identifying the resource for which the request is intended. The resource-URI does not contain a query string.

Beside SOAP, there is in prior art the above-mentioned second relevant Web Service standard:
The *Web Services Description Language* (WSDL) is an XML document for describing Web Services as a set of endpoint operations on messages containing either
document-oriented or Remote Procedure Call (RPC) payloads.
So called service interfaces are defined abstractly in terms of message structures and sequences of simple message exchanges (or "operations" in WSDL terminology). They are then bound to a concrete network protocol and data-encoding format to define an endpoint. Related concrete endpoints are bundled to define abstract endpoints (services).
WSDL supports a service interface definition that is distinct from the protocol bindings used for service invocation. WSDL allows for multiple bindings for a single service. The service interface definition and the access binding are also distinct from the implementation of the functionality of the service. Service requestors usually generate client stub code for a web

service from the corresponding WSDL; the WSDL of a service is usually requested from the service provider. The client stub code implements the necessary logic to create the correct message structure and the correct data encoding to address the endpoint. Since there is a distinction between definition, binding and implementation of a service, client stub codes created for a certain definition and binding can usually address various endpoints without requiring code changes, simply by using another endpoint address. **Fig. 5A** and the continuation thereof, **Fig. 5B** is given to disclose an exemplary WSDL document with further details to a person skilled in the art.

Having now described the constraints in which electronic communication of the above mentioned kind runs, the disadvantages of prior art will be described next below:

An important feature of Web Services is that they are stateless, according to a request-response scheme. A stateless server is one which treats each request as an independent transaction, unrelated to any previous request. This simplifies the server design because it does not need to allocate storage to deal with conversations in progress or worry about freeing it if a client dies in mid-transaction. A disadvantage is that it may be necessary to include more information in each request and this extra information will need to be interpreted by the server each time.

Such context information relative to such services is generally known in prior art. It may specify any additional qualification referring to the service itself, or the service requester or to the service provider, time, date, personal IDs, quality of service, etc. It is difficult to include contextual information into the communication between service requester and service provider other than by means of transporting it into said header fields as described with reference to fig. 2 and fig. 4.

Furthermore, in more elaborated architectures, web services are embedded in a web service management infrastructure. To serve this purpose, this management infrastructure may, in contrast to the web service itself, also require contextual information, also simply called context information.
As an example, a management component, which checks the authorization of a web service requestor, needs the requestor's identity in form of context information. This context information must not be visible in the web service interface, since it is independent from the web service itself and tied to the specific management infrastructure. Thus, in prior art the obvious solution to this would be to change the interface definition of the service and include an identifier. Since this adds a structural element to the message without specifying its value, this implies that it lies within the service requestor's responsibility to specify the correct value for this element. This, however, requires code changes in the client and in the service implementation as well and is not feasible in many situations.

1.3. OBJECTIVES OF THE INVENTION

It is thus an objective of the present invention to provide an improved method according to the preamble of claim 1 or 2, and a respective system.

2. SUMMARY AND ADVANTAGES OF THE INVENTION

This objective of the invention is achieved by the features stated in enclosed independent claims. Further advantageous arrangements and embodiments of the invention are set forth in the respective subclaims. Reference should now be made to the appended claims.

The term "endpoint" of a message denotes the resource (any kind of hardware or software resource) which either sends or receives a message. The term "endpoint" of a request denotes the resource which either submits the request or is intended to receive and execute the request. The term "endpoint specification" denotes the identifier of such resources. Any kind of endpoint specification representation is possible. Typically, an endpoint specification is given as HTTP-URL or request-URI.
Figure 1 contains the endpoint specification given as request-URI "/StockQuote".
In relation to Webservices, an endpoint is a resource, which either receives messages of a specific format or realizes a specific operation characterized by the specific format of the request and response messages, the specific data encoding and the specific network protocol. An endpoint typically is specified by a protocol-level endpoint address, e.g. an HTTP-URL or a request-URI.
By a "stateless" request is meant that a request is handled as separate transaction and is not related to any other (previous or future) request. Typically, a request is processed by the server and the response is returned immediately, then all information on the server related to this request is deleted. In other words, the server does not save the state of a request after the request is processed.

When further said context information is transported in a query string part of said endpoint specification, the handling of it is most easy.

The service requester is enabled according to the advantageous features of the present invention to specify the context for the web service request. Since the parameters specifying context information are part of the messages constituting service requests, the invention is well suited for (stateless) web services.

The present invention uses and leverages existing standard web service protocols. It does not require extensions of existing web service infrastructure and is transparent to both the web service infrastructure and the service implementation. It is therefore well suited for heterogeneous environments like ".NET" and Java web services).
The service interfaces are not changed, the messages still conform to the standards, and tools for building client proxies and client APIs can still be used without change and recompilation.

Further, in accordance to a very preferred use of the inventional method, said context information may advantageously comprise parameters specifying a contract between a respective service provider and a respective service requester.

Thus, when said context information relates to the conclusion of a particular contract between a service provider an a service requester in an environment comprising a multitude of contracts between them two, the managing of such services is improved by virtue of the feature that a particular contract may be specified easily by the requester.

Thus, in an e-business on demand infrastructure based on agreements between service provider and service requester, the present invention empowers the service requestor to control the usage conditions for service requests by including contract selection parameters as context information in the service request message. This is an important pre-condition for customer acceptance of web services, as costs for such web services become clear by that.

The present invention comprises thus advantageously a method which includes context information in a web service request in a way that

- the request message structure is not changed and compatibility with existing web service components is maintained
- the request message conforms to the current web service standards
- the context information is available to the components implementing this scheme
- Prior art components and generic components not implementing this scheme are not affected and the context information is transparent to them.

The present invention discloses further how to include context information in a web service description document in a way that

- the web service description document structure is not changed an still conforms to the current web service standards,
- the message structure is not changed,
- the context information only requires minimal client code changes or no client code changes at all,
- the context information can be personalized for specific identities. An identity may identify a real person or any component of a computer / software infrastructure like a computer program or a web service. The personalization process to create specific context information is therefore not limited to human users but comprises any other distinguishable entity including other web services.

The present invention further comprises advantageously to use context information in web service requests for selecting the usage conditions, which may be stored in the form of electronic contracts for web service requests.

Finally, the present invention defines a way that enables the client / service requestor to include context information in a service request by leveraging existing Web Service protocols and

infrastructures without changes. According to this preferred aspect of the invention the messages constituting a service request sent to the server by the client contain information, which is used by one or multiple components on the service provider side to establish the requested context.

3. BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and is not limited by the shape of the figures of the drawings in which:

Fig. 1 is a code section representation showing a prior art SOAP request message contained in a HTTP POST request message;

Fig. 2 is a code section representation showing the structure of a prior art HTTP request;

Fig. 3 is a code section representation showing a detail structure of a prior art SOAP message;

Fig. 4 is a code section representation showing the overall structure of a prior art SOAP message contained in a HTTP POST request message;

Fig. 5A is a code section representation showing a section of a prior art exemplary WSDL document used for describing web services;

Fig. 5B the continuation of Fig. 5A;

Fig. 6 is a block diagram representation illustrating the components of a web service infrastructure with indication of the control flow during the processing of a service request message according to prior art.

Fig. 7A is a code section representation showing a sample client application with code -see the arrow - to add context information to the endpoint address, according to a preferred aspect of the invention;

Fig. 7B is a code section representation showing a SOAP message carried by the HTTP POST command including context information according to the invention;

Fig. 8 is a pseudocode section representation showing a server

implementation for evaluating the context information according to the invention;

Fig. 9A is a pseudocode section representation showing a Web service description document (expressed in WSDL) aggregated with context information according to the present invention -see arrow;

Fig. 9B the continuation of Fig. 9A;

Fig. 10 is a block diagram representation showing the control flow and its relevant steps in a message exchange performed according to the present invention;

Fig. 11 is a block diagram representation according to fig. 10 and showing some pre-steps performed in an alternative message exchange performed in a preferred aspect of the present invention;

Fig. 12 is a block diagram representation showing the control flow and its relevant steps in a message exchange performed in a case when said context information is contract specifying information according to a preferred use of the present invention;

Fig. 13 is a block diagram representation showing the control flow and its relevant steps in a message exchange performed according to the basic approach provided by the present invention;

4. DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

With general reference to the figures and with special reference now to **fig. 6** service requests are realized in prior art by exchanging SOAP messages between a SOAP compliant client 60 and a respective Web server 62 server. The HTTP POST operation is used to send- step 610- the message from the client to the server and to receive - step 680- the result message. On the server side, the web server 62 or a servlet engine like Apache Tomcat and a SOAP server 64 like Apache AXIS realize the web service environment. Optionally the service provider

infrastructure may contain a web services management component 65. An implementation of a web service (e.g. a Java class) is deployed to this environment; the environment processes in steps 630, 640 650, 660, and 670 the service requests to the service implementation. Any of these components on the side of the service provider may be able to receive context information, but is not required to do so).

The client 60 comprises a Web service client application 61, a Java program containing a SOAP library like Apache AXIS and optionally comprising a proxy class generated from a web service description.

According to the invention, and with reference to **fig. 13**, illustrating the basic logic of the present invention, the context information is encoded by the client side 60, 61 as part of the endpoint specification of a service request, see step 1300. The endpoint is the address, where the deployed web service can be invoked on the server. It normally consists of a URL. The service request messages contain the specification of the endpoint.

The query string part does not affect the processing or routing of the message, since it is not considered for finding the service implementation by the servlet engine, the SOAP server or any prior art component. Thus, it is preferred to include the context information therein.

Then in client step 1305 the Soap request enriched by said context information, i.e., personal date like UserID= Bob, or further parameters, as mentioned above, is sent to the server side, see also step 610 in Fig. 6, showing prior art.

Then in step 1310 the server receives the request, receives by that the context information, step 1320, and evaluates -step 1330- the context information from the query string of the

request. By that the server can establish -step 1340- the context for the desired service and can invoke the service as detailed by the personal data and parameters from the above mentioned query string.

Further, in step 1350 the respective service is assembled, a respective SOAP response message is generated, step 1350, and sent to the client, which receives that message in step 1360.

Fig. 7A shows a sample client application 61 with code to add context information to the endpoint address. The sample client builds upon the Axis SOAP client for Java and uses the according Axis interfaces.

From **Fig. 7B** reveals an example for a resulting SOAP message carried by the HTTP POST command including context information; compared to Fig. 1, solely the endpoint address is changed according to the invention, see the arrow. It includes now the context information: "USERID = Bob".

The endpoint address is available to all components on the side of the service provider, either as part of the original message representation or through the different components APIs, but the query string part carrying the context information is ignored by prior art components.
The context information included in the service request is visible to those components, which implement this invention. Such a component extracts the context information from the endpoint address and uses it to establish the context for the service request.

Fig. 8 shows a pseudo code section for a server implementation for further improving clarity, how to evaluate the personal context data in the query string of the client request. The respective program code is part of the Web service management infrastructure and is advantageously implemented simply as add-

on to the prior art SOAP server 64 in Fig. 6. Advantageously, the code uses the usual prior art SOAP servers interfaces. As the pseudo codes can be directly understood by a person skilled in the art, a further description thereof need not be done in here.

According to the invention this scheme takes advantage of the following facts:

- Web service standards allow any endpoint address;
- The query string is available to all service provider components
- The query string is not used by the prior art service provider components
- Any context information can be encoded as string and added to the query string. The message containing this endpoint address therefore fully conforms to the SOAP standard.

Thus, it is to be understood that even more context information than a User ID alone can be transported in the query string, when for example some separation symbol as "+" or the like is defined in the client and the server component. Thus, also one or more parameters a, b, c, d, ... can also be transmitted in the query string of the request. For example, the query string can comprise a string like:

"USERID=Bob+a=2+b=3+c=15"

or

"a=2+b=3+c=someInformation"

by that additional personal information, the request can be detailed according to the present invention, for example in a situation, in which the client person wants to select some details out of a plenty of service offerings offered by the service provider. This accelerates and simplifies the whole procedure of service execution, as a complex task can be performed with a single stateless request.

This will be detailed next below.

According to a preferred aspect of the present invention, the following scheme to include context information in a web service description document builds upon the concepts explained so far:

Web service description documents (expressed in WSDL) are aggregated with context information according to the present invention. By selecting an appropriate web service description, a service requester can then select the web service AND a desired context, i.e. personal, customizing information, for the request. As an example, a service requester may have two WSDL documents for a service A, one for a performant but expensive implementation and one for a slower and cheaper implementation, and may choose from these WSDLs according to his current needs.

Depending on the kind of adaptation, the method according the present invention may change the parts of the WSDL document defining the structure of the message, the structure of the binding or data encoding, or it may change the value of message parameters, i.e. the endpoint address. The context information is encoded best in the query string part of the endpoint address according to the present invention. This is depicted in **fig. 9A** and the continuation thereof, **fig. 9B**.

According to the present invention, a preferred inventional scheme of including context information in a web service description document (WSDL) is depicted in **fig. 10**.

In fig. 10 again, steps performed by the client are depicted left, and those of the server side are depicted right. This scheme includes the following steps:

Client step 1010: Retrieving a pre-stored generic service description, or alternatively receiving a generic service description from a server.

Client step 1020: Adapting this service description by adding or modifying parameters constituting additional information, as mentioned shortly above.
This adaptation is done by step 1022, i.e., selecting these parameters dependent of the intended context, which represents thus the desired personalization of the request.
In step 1024 the adaptation of a service interface description document is done in such a way that only the implementation part (e.g. the endpoint address) is changed, while the definition and binding parts remain unchanged. This ensures that the SOAP message structures are not changed by the process performed according to the invention, and that the implementation of the invention requires minimal code changes or no code changes on the client at all.
In step 1026, the SOAP service request is prepared according to the adapted service description.
Then in client step 1030 the Soap request enriched by said personal date like UserID= Bob, or further parameters, as mentioned above, is sent to the server side, see also step 610 in Fig. 6, showing prior art.

Then in step 1040 the server receives the request and evaluates -step 1045 - the context information from the query string of the request. By that the server can establish the context for the desired service and can invoke the service as detailed by the personal data and parameters from the above mentioned query string.

Further, in step 1050 the respective service is assembled, a respective SOAP response message is generated and sent - step 1055 - to the client, which receives that message in step 1060.

A further scheme of personalization of a web service description document is depicted in **fig. 11**, which comprises some preceding steps compared to fig. 10:

In a step 1110 the client requests a WSDL document for a user whose identity is assumed to be given by the string variable "UserID = Bob". The request is sent in a step 1115.

In contrast to figure 10, the context information in the WSDL document is created now by the SOAP server 64 (fig. 6) specifically for the client identity of the service requester named Bob and sent back to the client in step 1125. Each so personalized WSDL is based on the same generic WSDL and is to be used by one client identity.
Step 1120 is explained in more detail by the following pseudo code section in which exemplary users "Bob" and "Billy" are referred to:

```
Begin pseudocode:
Read generic web service description document from disk
Parse xml structure
Get endpoint_base from parsed xml representation
if (user_id="Bob")
  String endpoint = endpoint_base + "?USERID=Bob";
else if (user_id="Billy")
  String endpoint = endpoint_base + "?USERID=Billy";
Change endpoint address in document
End pseudocode
```

The rest of the procedure is performed as described above with reference to fig. 10, step 1026. From the client point of view the usage of the Web Service is done just like any regular Web Service call. The client needs not to be aware of the personalized WSDL used to make the call. The SOAP request from all requestors using a personalized WSDL based on the same generic WSDL will be directed to the same servlet running on the same server. Because of the personalized enrichment of the endpoint, the server can determine the identity of the client.

As a person skilled in the art may appreciate, the inventional methods do not affect the definition and binding sections of the WSDL and do not change the message structure; the endpoint address is included in the corresponding message as string value. In consequence the structure of the client stub code does not need to be changed. If the endpoint address is read during runtime (e.g. from the WSDL document) and not fixed in the client code, there are no code changes at all.

To illustrate a further preferred use and embodiment of the invention, which can be built upon the concepts explained so far, we use the term "electronic contract" or "contract" to denote a representation of an agreement between two parties (e.g. service provider and service requestor) about the conditions for using web services or web applications. The contract details may specify the conditions for billing the services (e.g. price), service level agreements and further information that are highly critical for both the service requestor and the service provider.
There are no limitations regarding the scope of a contract or the number of contracts: one contract may contain multiple services and one service may be contained in multiple contracts, which are valid at the same time.

The electronic contracts are managed by appropriate web service management components, which can be deployed on the side of the service provider and on the side of the service requester.
In prior art, only web service management components on the side of the service provider are involved in managing contracts. Since a service can be contained in multiple contracts, these components also realize logic to select one contract from multiple contracts valid for this request before the request is processed. In prior art, the service request message does not contain contract related context information. Therefore the service requester can not select the contract to use for a service request or influence the process of selection. In other

words: The conditions being in effect for him are selected by a foreign party.

In a preferred application as depicted in **fig. 12**, i.e. use of the present invention, the service requester adds contract selection parameters as context information in the above sense to the service request message (SOAP) and sends this enriched request to the server. The request is received there, step 1220 - and the contract selection parameters are then extracted - step 1230 -from the request message by the service provider side program components and are analyzed and used during the process of contract selection, see step 1240. In addition to the evaluation of these contract selection parameters, the web service management components can implement further logic to select a specific contract for a service request. Then, in a step 1250 a respective SOAP message containing the desired particular service result is prepared by the server and sent to the client, who receives its desired particular response, step 1260.

The present invention can be realized in hardware, software, or a combination of hardware and software. A tool according to the present invention can be realized in a centralized fashion in one computer system, or in a distributed fashion where different elements are spread across several interconnected computer systems. Any kind of computer system or other apparatus adapted for carrying out the methods described herein is suited. A typical combination of hardware and software could be a general-purpose computer system with a computer program that, when being loaded and executed, controls the computer system such that it carries out the methods described herein.

The present invention can also be embedded in a computer program product, which comprises all the features enabling the implementation of the methods described herein, and which - when

loaded in a computer system - is able to carry out these methods.

Computer program means or computer program in the present context mean any expression, in any language, code or notation, of a set of instructions intended to cause a system having an information processing capability to perform a particular function either directly or after either or both of the following

a) conversion to another language, code or notation;

b) reproduction in a different material form.

C L A I M S

1. A method for processing context information in web based applications performing service requests and respective service provisions, in which method a web service can be invoked at an endpoint,
characterized by the step of:
a) receiving (1320; 800) context information specifying the context of the service request as a part of the endpoint specification of said request
b) evaluating (1330) said context information,
c) providing (1340) the service according to said evaluation.

2. A method for processing context information in web based applications performing service requests and respective service provisions, in which method a web service can be invoked at an endpoint,
characterized by the step of:
inserting (1300; 700) context information specifying the context of the service request as a part of the endpoint specification of said request,
c) issuing (1305) said request for said service via network,
d) receiving (1360) the service as defined by said context information.

3. The method according to claim 1 or 2, in which said context information is transported in a query string part of said endpoint specification of said request.

4. The method according to claim 1, in which said context information is included (1120) in said endpoint specification of a web service description document.

5. The method according to claim 2, in which said context information is extracted (1010, 1020) from a web service description document.

6. The method according to claim 1 or 2, wherein said context information specifies one or more items of the group:
a) an additional qualification referring to the service itself,
b) an additional qualification referring to the service requester,
c) an additional qualification referring to the service provider,
d) time,
e) date,
f) personal IDs.

7. The method according to claim 1 or 2, in which said context information comprises parameters specifying a particular contract between a respective service provider and a respective service requester.

8. A computer system having means for performing the steps of a method according to one of the preceding claims 1 to 7.

9. A computer program for execution in a network server system and for processing context information in web based applications performing service requests and respective service provisions, in which method a web service can be invoked at an endpoint, wherein said computer program comprises a functional component for performing the steps of:
a) receiving (1320; 800) context information specifying the context of the service request as a part of the endpoint specification of said request
b) evaluating (1330) said context information,
c) providing (1340) the service according to said

evaluation,
when said computer program code portions are executed on a computer.

10. A computer program for execution in a network client system and for processing context information in web based applications performing service requests and respective service provisions, in which method a web service can be invoked at an endpoint, wherein said computer program comprises a functional component for performing the steps of:
inserting (1300; 700) context information specifying the context of the service request as a part of the endpoint specification of said request,
c) issuing (1305) said request for said service via network,
d) receiving (1360) the service as defined by said context information,
when said computer program code portions are executed on a computer.

11. A computer program product stored on a computer usable medium comprising computer readable program means for causing a computer to perform the method of anyone of the claims 1 to 7,
when said computer program product is executed on a computer.

A B S T R A C T

The present invention refers to the field of networked computer telecommunication, and in particular to a method and system for processing context information in web based applications for providing Web Services via respective requests. In order to simplify such communication it is proposed to include (1300) context information into the endpoint specification of a service request. The advantage results that no changes to existing web service interfaces are necessary.
(Fig. 13)